

Towards Breaking the Curse of Dimensionality for High-Dimensional Privacy: An Extended Version

Hessam Zakerzadeh*

Charu C. Aggrawal†

Ken Barker‡

Abstract

The curse of dimensionality has remained a challenge for a wide variety of algorithms in data mining, clustering, classification and privacy. Recently, it was shown that an increasing dimensionality makes the data resistant to effective privacy. The theoretical results seem to suggest that the dimensionality curse is a fundamental barrier to privacy preservation. However, in practice, we show that some of the common properties of real data can be leveraged in order to greatly ameliorate the negative effects of the curse of dimensionality. In real data sets, many dimensions contain high levels of inter-attribute correlations. Such correlations enable the use of a process known as *vertical fragmentation* in order to decompose the data into vertical subsets of smaller dimensionality. An information-theoretic criterion of mutual information is used in the vertical decomposition process. This allows the use of an anonymization process, which is based on combining results from multiple independent fragments. We present a general approach which can be applied to the k -anonymity, ℓ -diversity, and t -closeness models. In the presence of inter-attribute correlations, such an approach continues to be much more robust in higher dimensionality, without losing accuracy. We present experimental results illustrating the effectiveness of the approach. This approach is resilient enough to prevent identity, attribute, and membership disclosure attack.

1 Introduction.

The problem of privacy-preservation has been studied extensively in recent years, because of the increasing amount of personal information which has become available in the context of a wide variety of applications. Starting with the seminal work in [1], a significant amount of work has been done on the problem of privacy preservation of different kinds of data. Numerous models [1, 5, 7, 18, 19, 22, 24] have been proposed for the problem of privacy preservation. However, it has been shown that data anonymization is increasingly difficult with dimensionality [2, 3], and the challenges extend to most privacy models.

The reason for the ineffectiveness of high-dimensional algorithms is simple. With increasing dimensionality, a

larger number of attributes are available for background attacks, even when the perturbation on a single attribute is significant. As a result, it has been shown theoretically in [2] that significantly larger perturbations are required with increasing dimensionality, and this reduces the effectiveness of the approach for privacy preservation. These results extend to a variety of models such as k -anonymity and ℓ -diversity[6].

An important observation about a *blind* anonymization process is that it often does not (fully) recognize that dependencies among the attributes may make a particular combination of dimensions more or less susceptible to anonymization. While such dependencies are *implicitly* utilized by many anonymization methods, their impact is often diluted by the overall anonymization procedure. Furthermore, the same dependencies impact the amount of information, which may be available in a particular subset of attributes for data mining applications. For example, an attribute such as *Age* and *Salary* may be highly correlated, and the *differential* impact of adding the attribute *Salary* may be less than adding another attribute such as *Sex* to the data.

One solution to the curse of dimensionality is to simply use feature selection [21, 23] in order to reduce the dimensionality of the data set, and retain a small subset of attributes which retains non-redundant information for a particular application. However, it is inevitable, that a pure feature selection approach will lose a significant amount of information for many application-specific scenarios. Therefore, a relevant question is as follows: “*Is it still somehow possible to retain all the attributes in the data, while using the non-redundancy of some subsets of attributes in the anonymization process to prevent identity and attribute disclosure attack, and also retain most of the utility in the data for mining scenarios?*”.

A less drastic approach than feature selection is the concept of *vertical fragmentation*. The idea is to break up the data set into different subsets of attributes using vertical fragmentation, and anonymize each subset independently. The results from the different subsets of attributes then need to be combined for a particular application. Since all attributes are still retained, the amount of information loss of fragmentation is less than that of feature selection. The exact nature of the fragmentation may depend upon the specific

*University of Calgary, hzakerza@ucalgary.ca

†IBM T.J. Watson Research Center, charu@us.ibm.com

‡University of Calgary, kbarker@ucalgary.ca

application at hand. For example:

- In a supervised application, the fragments may be completely disjoint and share no attributes other than the class attribute. Thus, while the correspondence information among different fragments is lost, this may not be as critical, since the class variable can be independently learned from each fragment. The amount of information lost is limited in such cases, especially if the individual fragments are carefully chosen based on information-theoretic principles. The results from the different fragments can then be combined carefully on an *aggregate basis* in order to obtain high quality classification results. Care needs to be taken in the fragmentation process that the common class attribute may not be used in order to partially join the fragments together, and reduce the anonymity.
- In an unsupervised application, the fragments may need to have one or more common attributes in order to ensure a limited level of correspondence between different fragments. This case is actually not too different from the supervised case. The main difference is that instead of the class attribute, it is the common attribute which needs to be carefully accounted for during the fragmentation process.

In this paper, we primarily focus on the supervised scenario of classification as a first application. The generalizations to other unsupervised scenarios will be handled in future work.

It should be emphasized that while the *theoretical* results of the dimensionality curse still hold true [2], their *practical* impact can be greatly alleviated by carefully accounting for the nature of the data set in a particular application. Pathological cases may exist in which every feature is independent of one another, and in such cases, the earlier theoretical results on the curse of dimensionality continue to hold true. However, such pathological cases rarely arise in practice. Therefore, the goal of fragmentation is to leverage the mutual information within different features in order to alleviate the dimensionality curse in the vast majority of cases. The experimental results of this paper show that the fragmentation method can achieve significant improvements over the currently available methods. It should also be emphasized that the fragmentation method is a *meta-algorithm* which can be *combined with any existing anonymization algorithm* in order to improve its effectiveness. We start with the k -anonymity model in this paper because we believe such a hard problem like curse of dimensionality must be first addressed in the simplest and most relaxed privacy model. Then, we explain how the fragmentation process can be generalized to satisfy the ℓ -diversity[22] (or t -closeness[19]) requirement. In addition, we discuss how the fragmentation brings in the membership disclosure protection[32]. In general, the fragmentation process has the

potential to be extended for other privacy models, because of its meta-approach, which is more easily generalizable. This might eventually provide unprecedented flexibility in using the fragmentation method as a general-purpose meta-algorithm in the context of a wide variety of scenarios.

This paper is organized as follows. The remainder of this section discusses related work. Section 2 discusses the overview of the approach for the k -anonymity. Section 3 discusses details of the fragment-based k -anonymization method. Extending the fragmentation approach for ℓ -diversity (or t -closeness) is shown in Section 4. In addition, this section shows how the fragmentation-based anonymization can prevent the membership attack. The experimental results are presented in Section 5. Section 6 contains the conclusions and summary.

2 Related Work.

The problem of privacy preservation was first studied in [1]. This approach was based on noise-based perturbation of the underlying data. Subsequently, the problem of k -anonymization of data was proposed in [24]. Other models for data privacy have been proposed in [22, 19]. Numerous methods [15, 16, 11, 17, 27, 28] have been proposed for effective and efficient k -anonymization, and this continues to remain one of the most widely used models in the literature because of its simplicity. The theoretical results illustrating the degradation of privacy-preservation methods with increasing dimensionality have been discussed in [2, 3, 4]. The work in [12] proposed an anonymization method for high-dimensional data. However, it is only applicable to sparse transactional data, and is heavily dependent of the sparse structure of transaction data in order to achieve this goal. Kifer [33] suggested the idea of releasing anonymized marginals beside the anonymized original table. However, the original table must still be anonymized as a whole which results in high information loss. Furthermore, publishing the marginals makes the published data highly workload dependant. The privacy models in [29, 30, 31] can cope with the problem of curse of dimensionality by reducing the number of quasi-identifiers¹. That is, they make an assumption about the number of quasi-identifiers known by an attacker, and apply anonymization only on limited number of quasi-identifiers. However, this assumption may not hold true in some cases. Other relevant works [26, 20] are able to provide some protection in the high dimensional case, though they can be challenged in some circumstances [14]. Furthermore, any of these methods can be used in combination with our approach, which is designed as a more general purpose meta-algorithm. Finally, the concept of vertical partitioning and fragmentation has been explored in the context of dis-

¹Although the works in [29, 30] are originally proposed for the set-valued (transaction) data, the relational data can be transformed to set-valued data for anonymization.

tributed data privacy with cryptographic protocols [25], or for capturing confidential privacy constraints in the context of such methods [8, 10]. The goals and motivations of these methods are quite different, and are not at all focused on the problem of high dimensional anonymization.

In the context of increasing dimensionality, it is natural to explore feature selection [23, 21] as an alternative in order to reduce the data dimensionality. However, this is a rather drastic solution, which can impact the quality of the underlying results significantly. Therefore, this paper proposes the approach of fragmentation as a general-purpose meta-algorithm in order to improve the robustness of high-dimensional anonymization algorithms.

3 Overview of The Approach for The Fragmentation K -Anonymity

In this section, we first introduce the most important notations used in the fragmentation-based k -anonymity in Table 1. An overview is then provided about vertical fragmentation, and its incorporation as a general-purpose meta-algorithm for privacy preservation.

Table 1: List of notations in fragmentation k -anonymity

notation	explanation
\mathcal{F}	a vertical fragmentation
F_i	i^{th} fragment in fragmentation \mathcal{F}
EQ_{ij}	j^{th} equivalence class in fragments F_i
C_{ij}	set of all class values in equivalence class EQ_{ij}
$\mathcal{P}(S)$	power set of set S
$ \cdot $	size of a set

3.1 Vertical Fragmentation Let T be a relation defined over a schema $T(A_1^f, A_2^f, \dots, A_n^f, A^c)$ where A_i^f represents the feature attributes, and A^c is the class attribute. A vertical fragmentation of relation T splits the feature variables into multiple non-overlapping fragments. Formally, a vertical fragmentation is defined as follows:

DEFINITION 3.1. (VERTICAL FRAGMENTATION). *Given a relation schema T , a vertical fragmentation \mathcal{F} of T is a partitioning of the attributes into fragments $\mathcal{F} = \{F_1, F_2, \dots, F_m\}$ such that each F_i contains a disjoint subset of the feature attributes. Therefore, it is the case that $\forall F_i \in \mathcal{F}, F_i \subseteq T$ and $F_i \cap F_j = \emptyset$ ($i \neq j$) and $\bigcup F_i = T$ ($i=1, \dots, m$).*

In addition, it is *implicitly assumed* that the class attribute A^c is associated with each fragment. As we will see later, the presence of this common attribute needs to be accounted for in a special way since it allows the reconstruction of *some* correspondence between the attribute values of different fragments. Therefore, methods need to be designed to ensure that this correspondence cannot be used in order to attack the anonymity of the fragmented data. In some cases, this process requires the perturbation of a few class values, in order to ensure non-identifiability.

Note that the fragmentation process is used as a meta-approach in conjunction with an off-the-shelf anonymization algorithm. A fragmentation \mathcal{F} is referred to as a *k-anonymous fragmentation* after applying an anonymization algorithm to it, if and only if the following two conditions hold:

- **Fragment k -anonymity condition:** Each fragment in \mathcal{F} satisfies the k -anonymity condition. This condition can be easily satisfied by applying any off-the-shelf k -anonymity algorithm to each fragment.
- **K -Anonymity non-reconstructability condition:** The relation resulting from joining any arbitrary fragments on the class variable satisfies the k -anonymity condition.

The number of possible fragmentations of a set of features is rather large in the high-dimensional case. For example, for a set of n features, there may be $O(n^n)$ possible fragmentations. Clearly, exhaustive search through all the possible fragmentations for the high-dimensional case may become prohibitive. Therefore, a systematic approach is required to search the space of possible fragmentations. Since this work is focussed on the classification problem, the fragmentation approach should attempt to maximize the amount of *non-redundant* information contained in each feature of a particular fragment, which is relevant for the classification process. Correspondingly, our systematic search approach utilizes a metric referred to as *Fragmentation Minimum Redundancy Maximum Relevance (FMRMR)* in order to create fragments.

3.2 Fragmentation Minimum Redundancy Maximum Relevance The ideal fragmentation is one in which the set of attributes in each fragment is a comprehensive representation of the information required for the mining process. Since this paper addresses the classification problem, the metric will explicitly use the class variable for quantification, though it is conceivable that the metric for other applications would be different. In the supervised context, a comprehensive representation refers to high predictability of the class variable from the features in each fragment, while minimizing redundancy. It is evident that the *simultaneous* incorporation of features with high mutual information within a given fragment does not provide any additional advantages, even when they are all highly relevant to the class attribute. This implies that a combination of the relevance to the class attribute and the mutual information with respect to one another can be useful for the process of constructing a fragment.

To this effect, we draw on the feature selection literature, which defines the concept of the *Minimum Redundancy Maximum Relevance (MRMR)* metric[9, 23]. This metric uses a dependency quantification (denoted as W) among the feature variables and a dependency quantification (denoted as V) between the feature variables and the class attribute in each fragment. Our proposed heuristic aims at maximizing

the summation of MRMR for all the fragments in a given fragmentation. The Fragmentation MRMR (FMRMR) is the summation of the values of MRMR within a fragment. This value is defined for a particular fragmentation \mathcal{F} as follows:

$$FMRMR(\mathcal{F}) = \sum_{t=1}^{|\mathcal{F}|} (V_t - W_t)$$

$$V_t = \frac{1}{|A_t^f|} \sum_{j \in A_t^f} I(cls, j)$$

$$W_t = \frac{1}{|A_t^f|^2} \sum_{k, j \in A_t^f} I(k, j) \text{ where:}$$

- A_t^f : set of features in fragment t of fragmentation \mathcal{F}
- $I(x, y)$: mutual information between attributes x and y
- V_t : total mutual information between the features and the class attribute in fragment t of fragmentation \mathcal{F}
- W_t : total pairwise mutual information between the features in fragment t of fragmentation \mathcal{F}
- cls : the class attribute

The overall approach for the k -anonymity uses a three-step technique for the fragmentation process. For a high-dimensional relation T with n features and one class attribute, these three steps are as follows:

1. Use a carefully-designed search algorithm to decompose the relation into fragments. The constructed fragments have non-overlapping sets of features together with the class attribute. The fragmentation process uses the afore-mentioned measure in order to determine the optimal fragments.
2. Anonymize each fragment separately using an existing anonymization algorithm, such as the Mondrian multi-dimensional k -anonymity algorithm [15].
3. At this point, it should be noted that the anonymized fragments can be (partially) joined back using the common attribute, which in the supervised scenario is the class attribute. Depending on the distribution of values in the common attribute, the result might violate the k -anonymity constraint. This is essentially a *k -anonymity non-reconstructability condition violation*. Therefore, additional steps are required in order to ensure non-reconstructability. The techniques for achieving this are slightly involved and distort the class variable in such a way that non-reconstructability is guaranteed. These methods will be described in the next section. It should be noted that the distortion of the class variable may result in some further reduction in accuracy. However, in *practice*, for most reasonable distributions, the required distortions are very limited, if any.

The second step in the afore-mentioned list does not require further explanation. Therefore, the exposition in this paper will describe the detailed methods for performing the first and the third steps. For the third step, three different alternatives will be proposed. It should also be noted that although the class and sensitive attributes have been

considered the same in many works, they might be different, and data contains many sensitive attributes in practice. In such cases, the other sensitive attributes also need to be fragmented in order to ensure that the two fragments cannot be joined. However, they should be fragmented only *after* the quasi-identifiers have already been fragmented (using the same approach as discussed in the next section). This is because it is more critical to ensure that quasi-identifiers are evenly distributed among fragments. Therefore, what follows will only focus on quasi-identifiers for simplicity.

4 Fragmentation-based K -Anonymization

In this section, the first and third steps in the afore-mentioned fragmentation meta-algorithm will be discussed. First, the fragment construction heuristic will be introduced.

4.1 Fragment Construction Heuristic As the number of features increases, the number of possible fragments grows exponentially. This explosion in the number of fragments makes exhaustive search in this space impractical. Therefore, we propose an algorithm which tries to form a fragmentation with maximum $FMRMR$. For simplicity, a binary fragmentation into two parts will be described, though it is possible in principle to fragment into multiple parts by repeating the process.

We define the *FMRMR contribution* of a feature attribute A_i^f with respect to fragment F_j of fragmentation \mathcal{F} as the difference between $FMRMR$ of \mathcal{F} after and before adding A_i^f to F_j . The quantification of the mutual information between the n features and the class attribute is stored in the form of an $(n + 1) \times (n + 1)$ matrix denoted by $[MI]_{(n+1) \times (n+1)}$.

The *FMRMR* metric attempts not to place features having high mutual information in one fragment. Therefore, as a starting point, two features having the highest mutual information are picked as seeds and placed in different fragments. Afterwards, in a greedy manner, and while there exists un-assigned features, the *FMRMR contributions* of all unassigned features with regards to both fragments are calculated. The unassigned feature with the highest *FMRMR contribution* is added to the relevant partition. Finally, the common attribute (class attribute in the supervised case) is added to each fragment separately. The overall approach is illustrated in Algorithm 1 in the Appendix A in the supplementary materials.

4.2 The Final Step: K -Anonymity non-Reconstructability As indicated earlier, applying a k -anonymity algorithm on each fragment in order to satisfy the *fragment k -anonymity condition* is not sufficient for ensuring non-identifiability. This is because the common attribute (class attribute) can be used for (very approximate) joins, and such joins provide some additional information about fragment correspondence. Therefore, in theory, it may be possible that the overall anonymity level of the relation resulted from

joining k -anonymized fragments is less than k , though in practice it is rather unlikely because of the approximate nature of the join.

We call a fragmentation in which all fragments satisfy the k -anonymity condition *k -anonymity non-reconstructible* if the relations resulting from joining any arbitrary fragments on the class attribute satisfy k -anonymity. Similarly, a fragmentation is called *reconstructible* if k -anonymity is violated after joining some of its fragments. Definition 4.1 formally defines a *k -anonymity non-reconstructible* fragmentation.

DEFINITION 4.1. (K -ANONYMITY NON-RECONSTRUCTIBLE FRAGMENTATION). Fragmentation $\mathcal{F} = \{F_1, F_2, \dots, F_m\}$ which satisfies the fragment k -anonymity condition is called *k -anonymity non-reconstructible* if and only if $\forall s \in \mathcal{P}(\mathcal{F})$, the relation resulting from joining members (fragments) of s satisfies the k -anonymity condition.

The power set $\mathcal{P}(\mathcal{F})$ has 2^m members. However, the k -anonymity condition must be checked for members of size at least 2.

It should be noted that the joining process is only approximate and noisy, which is good for anonymization. Therefore, successful violation attacks of the type discussed above are often difficult to perform in practice. For example, joining will result in some tuples that do not have any corresponding tuple in the original table. These tuples are called *fake tuples*, and may sometimes be helpful for obfuscation of identification of relevant tuple identities.

As articulated in Definition 4.1, given a fragmentation \mathcal{F} with m fragments, the k -anonymity condition must be satisfied for $(2^m - m - 1)$ possible relations resulted from joining arbitrary fragments. However, the relations resulted from joining more than two fragments can be simply obtained by consecutive binary joins. This paves the way to define the k -anonymity non-reconstructability condition by joining only two fragments. For instance, checking the k -anonymity non-reconstructability condition on a relation resulting from joining members of $\{F_1, F_2, \dots, F_n\} \in \mathcal{P}(\mathcal{F})$ can be accomplished by checking the k -anonymity non-reconstructability condition in each of the following binary joins: $I_1 = F_1 \bowtie F_2, I_2 = I_1 \bowtie F_3, \dots, I_{n-1} = I_{n-2} \bowtie F_n$.

Therefore, for simplicity, we can continue our discussion with only two fragments, without any loss of generality. This condition is formally stated in Theorem 4.1. An important concept to continue with the remaining of the paper is to understand the notion of equivalence classes resulting from the anonymization. In the anonymized table, records with the same value for their quasi-identifiers constitute an *equivalence class*.

THEOREM 4.1. (K -ANONYMITY NON-RECONSTRUCTABILITY CONDITION). The condition

for a given fragmentation $\mathcal{F} = \{F_1, F_2\}$ which satisfies the fragment k -anonymity condition, and fragments $F_1 = \{EQ_{11}, EQ_{12}, \dots, EQ_{1n}\}$ and $F_2 = \{EQ_{21}, EQ_{22}, \dots, EQ_{2m}\}$, to be non-reconstructible is that one of the following must be true for each joined pair EQ_{1i}, EQ_{2j} :

- $\sum_{c \in C_{1i} \cap C_{2j}} \text{freq}(c, EQ_{1i}) \times \text{freq}(c, EQ_{2j}) = 0$
- $\sum_{c \in C_{1i} \cap C_{2j}} \text{freq}(c, EQ_{1i}) \times \text{freq}(c, EQ_{2j}) \geq k$

The proof of this theorem is presented in Appendix B in the supplementary materials.

Enforcing and satisfying the k -anonymity non-reconstructability condition in a fragmentation may require some of the class values to be distorted. As explained shortly, the change in the class values can be performed using various strategies. However, minimizing the number of changes is always desirable in order to retain accuracy. The design of an algorithm which provably minimizes the changes is computationally intractable because of the exponential number of possibilities. Therefore, we propose three heuristic strategies to enforce the k -anonymity non-reconstructability condition. It is worth mentioning that the utility of each strategy is different from others. Before providing a more detailed exposition, we introduce the concept of a *dependency graph*, which provides the logical construct necessary for a good algorithmic design.

4.2.1 Dependency Graph A *dependency graph* is an undirected graph structure which captures the dependency among different equivalence classes in a given fragmentation. Nodes in the *dependency graph* are equivalence classes, and there exists an edge between node EQ_{ix} and EQ_{jy} provided that:

1. $i \neq j$ that means EQ_{ix} and EQ_{jy} belong to two different fragments.
2. $C_{ix} \cap C_{jy} \neq \emptyset$ that means EQ_{ix} and EQ_{jy} have at least one class value in common.

The set of all equivalence classes in a fragmentation may be divided into subsets having no dependency on each other. In other words, no equivalence class from one subset can be joined with equivalence classes in the other subset. Thus, the *dependency graph* is not connected, and the *dependency graph* components reflect the full dependencies among all equivalence classes, rather than a single connected *dependency graph*. The process of constructing the *dependency graph* is shown in Algorithm 2 in Appendix A in the supplementary materials.

The k -anonymity non-reconstructability condition is enforced on each connected component of the *dependency graph* separately, since there is no inter-component dependency. The k -anonymity non-reconstructability condition on each connected component can also be achieved by enforcing it on each edge. We introduce three different

strategies in order to achieve this goal.

4.2.2 Naive Enforcement Satisfying the k -anonymity non-reconstructability condition for an edge between equivalence classes EQ_{1i} and EQ_{2j} can simply be done by enforcing each equivalence class to have only one class value. In the naive k -anonymity non-reconstructability enforcement approach, class values in each equivalence class are changed to the majority class in that equivalence class. In this case, two given equivalence classes either cannot be joined, or their join generates at least k^2 tuples. Such an approach is clearly suboptimal, and fails to take full advantage of the flexibility associated with distorting the class variable in a way which is sensitive to the behavior of the remaining data.

4.2.3 Dependency Graph-based Enforcement Unlike the naive approach, the class values in *only* equivalence classes violating the k -anonymity after being joined are changed to the majority class value in this approach. Another difference between this approach and the naive approach is that the dependency graph-based approach aims at minimizing the number of changes in each equivalence class. In order to achieve this goal, this approach changes only one attribute in each step.

Starting from a random node (*current-node*) in the *dependency graph*, the *dependency graph* is explored in a breadth-first manner. The k -anonymity non-reconstructability condition is checked between *current-node* and every single unvisited neighbor nodes. If the condition does not hold between *current-node* and one of its unvisited neighbors, the class values with lowest frequency in the neighbor node is changed to the majority class value until the condition is satisfied. After satisfying k -anonymity non-reconstructability between *current-node* and all its neighbor nodes, *current-node* is marked *visited*. This process must be repeated for all *components* in the *dependency graph* until all nodes are marked *visited*. The pseudocode of this algorithm is demonstrated in Algorithm 3 in Appendix A in the supplementary materials.

4.2.4 Enforcement via δ -selectivity In spite of the approximate nature of the join between different fragments, they are a potential threat to k -anonymity. Thus, the prevention of violating joins is important. Publishing the class values for *each and every* single tuple (row) in the anonymized fragment is a major cause of this violation.

The δ -selectivity approach changes the way in which class values are published. This enables a more relaxed k -anonymity non-reconstructability condition enforcement on the equivalence classes. Instead of publishing the class values on a *per tuple* basis, they are published on a *per equivalence class* basis with the use of *ambiguous values (slots)*. In an equivalence class, each class value has equal probability of being assigned to a tuple. This results in the possibility of assuming different instantiations (or

versions) for a given equivalence class. Then, given two equivalence classes, there exist multiple ways to join them, corresponding to different assignments of class values to tuples. The modified k -anonymity non-reconstructability condition leverages this ambiguity effectively. Appendix C.I in the supplementary materials exemplifies tuple-level and equivalence class-level class value publishing.

As mentioned above, the ambiguous slots in an equivalence class EQ_{ij} , published at the equivalence class-level, can take any of the class values in C_{ij} . In other words, different versions for EQ_{ij} can be assumed.

DEFINITION 4.2. (EQUIVALENCE CLASS VERSION). *An arbitrary assignment of class values available in an equivalence class EQ_{ij} to ambiguous slots in EQ_{ij} generates a version of EQ_{ij} shown by $V(EQ_{ij})$.*

Although publishing the class values at the equivalence class-level reduces the risk of k -anonymity violation, the resulting equivalence classes are still vulnerable to be joined back and violate the k -anonymity. As an example, consider the extreme case where the class values are unique in an equivalence class. Each tuple in the equivalence class is assigned to a different class value which is similar to the case in which class values are released at the tuple level. Given two equivalence classes whose class values are published in equivalence class-level and have at least one class value in common, there exist different ways to join them. The number of tuples resulted from joining two equivalence classes are referred to as *equijoin selectivity*. Among all possible joins, those generating k tuples (or more) are referred to as *k -anonymity-preserving equijoins*.

DEFINITION 4.3. (K-ANONYMITY-PRESERVING EQUIJOIN). *Given two equivalence classes EQ_{1i} and EQ_{2j} whose class values are published at the equivalence class level, and which share at least one class value, the join between $V(EQ_{1i})$ and $V(EQ_{2j})$ is a k -anonymity-preserving equijoin if and only if it produces at least k tuples.*

We can now define the equijoin selectivity privacy level in terms of the possible equijoins between two equivalence classes.

DEFINITION 4.4. (EQUIJOIN SELECTIVITY PRIVACY LEVEL). *The ratio of number of k -anonymity-preserving equijoins in joining two equivalence classes EQ_{1i} and EQ_{2j} to the total number of possible equijoins in joining the same equivalence classes is referred to as equijoin selectivity privacy level of EQ_{1i} and EQ_{2j} . This value is denoted by $\eta(EQ_{1i}, EQ_{2j})$. In other words, $\eta(EQ_{1i}, EQ_{2j}) =$*

$$\frac{|ds_p = \{ \{V(EQ_{1i})\} \bowtie \{V(EQ_{2j})\} \mid |\{V(EQ_{1i})\} \bowtie \{V(EQ_{2j})\}| \geq k \}|}{|ds_w = \{ \{V(EQ_{1i})\} \bowtie \{V(EQ_{2j})\} \}|}$$

Intuitively, $\eta(EQ_{1i}, EQ_{2j})$ indicates the probability that the result of joining EQ_{1i} and EQ_{2j} is a k -anonymous equivalence class. As an example, when the value of η is 1,

it indicates that $\forall v_1 \in V(EQ_{1i}), v_2 \in V(EQ_{2j})$, we have $|v_1 \bowtie v_2| \geq k$. In other words, all possible instantiations will result in a k -anonymity-preserving equijoin.

DEFINITION 4.5. (δ -SELECTIVE K -ANONYMOUS FRAG.). A fragmentation $\mathcal{F} = \{F_1, F_2\}$ that satisfies the fragment k -anonymity condition is called δ -selective if and only if $\forall EQ_{1i}, EQ_{2j}$ we have $\eta(EQ_{1i}, EQ_{2j}) \geq \delta$.

Algorithm 4 in Appendix A in the supplementary materials shows how δ -selectivity can be enforced on a fragmentation \mathcal{F} .

5 Extension to ℓ -Diversity

Analogous to the k -anonymity case, a fragmentation is called an ℓ -diverse fragmentation, if and only if 1) each fragment satisfies the ℓ -diversity requirement (fragment ℓ -diversity condition) and 2) joining the fragments does not violate the ℓ -diversity requirement (ℓ -diversity non-reconstructability condition). Satisfying the fragment ℓ -diversity condition is similar to that of the k -anonymity case. However, for ℓ -diversity, the non-reconstructability condition is different. Table 2 describes the notations used in this section. A brief overview of the steps for the fragmentation-based ℓ -diversity is provided below:

Table 2: List of notations in fragmentation ℓ -diversity

notation	explanation
\mathcal{F}	a vertical fragmentation
F_i	i^{th} fragment in fragmentation \mathcal{F}
S_i	i^{th} segment
CK_{ij}	data chunk belongs to segment S_i and fragment F_j
EQ_{ij}^s	i th equivalence class belongs to chunk CK_{sj}
C_{ij}	set of class values for CK_{ij}
C_{ij}^s	set of class values for EQ_{ij}^s
l_i	diversity level of segment S_i
$ \cdot $	size of a set

1. Use the fragment construction algorithm proposed in Section 4.1 solely to *compute* the best fragmentation $\mathcal{F} = \{F_1, F_2, \dots, F_n\}$. However, the fragmentation is not actually *executed* in this step.
2. Cluster data records into m segments $\mathcal{S} = \{S_1, S_2, \dots, S_m\}$ using a top-down clustering algorithm. Stop dividing each segment S_i into further sub-segments if the resulting sub-segments either violate the k -anonymity or ℓ -diversity requirement. Final segment S_i has the diversity level l_i .
3. Vertically partition each segment S_i , using the fragments found in Step 1, into n data chunks. The diversity level of each data chunk CK_{ij} is equal to the diversity level of segment S_i , which is l_i .
4. Use any off-the-shell ℓ -diversity algorithm to anonymize each chunk CK_{ij} . However, the diversity requirement of each CK_{ij} must be set to l_i .
5. Merge equivalence classes belonging to the same vertical fragments and publish them as one fragment.

In the afore-mentioned algorithm, it is worth noting that the clustering algorithm in the second step should be adjusted to the relevant workload. Since the workload in our approach is classification, a classification-oriented clustering algorithm results in higher utility.

The ℓ -diversity non-reconstructability condition may be violated if at least two chunks CK_{si} and CK_{sj} are joined and the resulting data set violates either the k -anonymity or the ℓ -diversity requirement. We prove that this will not be the case according to the way equivalence classes in chunks have been formed. For simplicity, we prove this for the case of two vertical fragments. However, the result is true in general.

THEOREM 5.1. *The data set resulting from joining chunks CK_{s1} and CK_{s2} neither violates k -anonymity nor ℓ -diversity.*

Again, the proof of this theorem is presented in Appendix B in the supplementary materials for the sake of brevity. It is easy to show that this extension can be utilized for t -closeness by simply enforcing t -closeness instead of ℓ -diversity in the aforementioned steps.

5.1 Membership Disclosure Protection Fragmenting the data can help protect against membership attack[32] by disassociating different attributes. As discussed in [32], the ability to determine presence or absence of a subject's record in the published data is a privacy threat. This can be done by comparing the subject's quasi-identifiers with the published quasi-identifiers.

Consider an attacker trying to find out the membership of subject v in the published fragmented data. As attributes are fragmented, the attacker must find the matching equivalence class in each fragment to which the subject's attributes belong. This may not be possible considering the generalization applied on the attributes. However, provided that the attacker succeeds in finding the matching equivalence classes $EQ_{1i}, EQ_{2j}, \dots, EQ_{np}$, the likelihood that the record pertaining to v exists in the published fragmented data is $\frac{|EQ_{1i} \bowtie EQ_{2j} \bowtie \dots \bowtie EQ_{np}|}{|F_1 \bowtie F_2 \bowtie \dots \bowtie F_n|} =$

$\frac{|EQ_{1i} \bowtie EQ_{2j} \bowtie \dots \bowtie EQ_{np}|}{\sum_i \sum_j \dots \sum_p |EQ_{1i} \bowtie EQ_{2j} \bowtie \dots \bowtie EQ_{np}|}$. This likelihood is mostly impacted by p (the number of vertical fragments), anonymity level (either k or ℓ), and $|D|$ (size of the data set). In most cases, the value of p is small, and $|D| \gg k$. Therefore, the numerator of the likelihood formula becomes much smaller than the denominator. When the value of p increases, the number of common class values among specific equivalence classes EQ_{1i}, EQ_{2j}, \dots , and EQ_{np} drops and even tends to zero in many cases. Therefore, the chance of a successful membership attack becomes negligible. In general, the fragmentation-based anonymization provides strong protection against membership attack.

6 Experimental Results

In this section, we will present the experimental results showing the effectiveness of our method. The goal is to

show that the fragmentation process is able to retain greater utility of the data both in terms of classification measures and information loss measures, at the same level of privacy.

We utilized two metrics to evaluate the effectiveness of our proposed method, *information loss* and *weighted F-measure* to capture the total amount of lost information and evaluate the utility of the data anonymized by our meta-algorithm, respectively. Details on how these metrics are used on fragmented data are available in Appendix D.I in the supplementary materials.

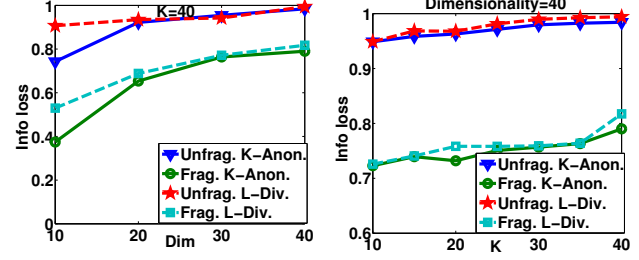
6.1 Baselines Since the goal was to show the effectiveness of the fragmentation approach as a meta-algorithm, the baseline for the approach were the results for the anonymization process with and without fragmentation. The Mondrian multidimensional anonymity method [15] was used for *both* the fragmented, *and* the unfragmented scenario. More accurately, in Step 2 of fragmented k -anonymity, we used *median* Mondrian and in Step 4 of fragmented ℓ -diversity, we utilized ℓ -diversity Mondrian. Therefore, the qualitative improvements show the effects of fragmentation, as a methodology to improve the effectiveness of an off-the-shelf approach.

6.2 Data Sets Real data set *Musk* from the *UCI Machine Learning Repository*² was used. The detailed description of the data set is provided in the Appendix D.II of the supplementary materials.

6.3 Results In each case, the results were measured with varying dimensionality and anonymity level. In each case, the anonymity level was varied after fixing the dimensionality, and the dimensionality was varied on fixing the anonymity level. The anonymity level was fixed to 40, when the dimensionality was varied on the X -axis. While varying the anonymity level on the X -axis, the dimensionality was fixed to 40. In case of ℓ -diversity, the value of ℓ is set to 2. In addition, we fixed δ to 0.5 in δ -selective enforcement approach. It is important to note that the information loss results do *not* vary with the different strategies for ensuring k -anonymity non-reconstructability, which affect only the class variable. Since the information loss metrics are based on the feature variables only, a single chart will be shown for the case of information loss, whereas the performance results for different approaches of k -anonymity (based on different strategies for ensuring k -anonymity non-reconstructability) and ℓ -diversity will be shown in the case of F-measure separately by means of solid and dashed lines, respectively.

The information loss with varying dimensionality is illustrated in Figure 1a. The dimensionality is illustrated on the X -axis, and the information loss is illustrated on the Y -axis in each case. Besides, diversity level is set to 2 for ℓ -diversity in all experiments. It is evident that the information

Figure 1: Information loss vs. dimensionality & k
(a) Info loss vs. dim (b) Info loss vs. k



loss of the *unfragmented* approach (for both k -anonymity and ℓ -diversity) increases with increasing dimensionality, which is in agreement of the results found earlier in [2]. In fact, the error touches almost its upper bound, which implies that each generalized value starts losing more and more of its specificity in the unfragmented case. On the other hand, the fragmentation method shows drastic improvements in the amount of information loss, for both k -anonymity and ℓ -diversity. This implies that a significant amount of attribute specificity is retained in each fragment.

The information loss with increasing anonymity level is illustrated in Figure 1b. The anonymity level is illustrated on the X -axis, whereas the information level is illustrated on the Y -axis. It is not surprising that the information loss increases with the anonymity level, and enforcing diversity. However, as in the case of the results with increasing dimensionality, the improvements achieved by fragmentation were significant.

The comparisons for the F-measure with increasing dimensionality are much more tricky. This is because the addition of more dimensions to a data set affects the classification precision and recall (and hence F-measure) of the data in two mutually contradictory ways:

- A larger number of dimensions provides greater knowledge (in terms of more attributes) to the classifier in order to improve its precision and recall.
- Data sets of larger dimensionality will have greater information loss on a *per attribute basis*, and this reduces the effectiveness of the classifiers.

So how does this tug-of-war between two mutually contradictory effects impact the final classification results, and how does the fragmentation process affect this tradeoff? Figure 2 compares the prediction F-measure of the unfragmented and fragmented anonymization methods with increasing dimensionality for two classifiers. In all cases, the different variants of the fragmentation scheme have higher classification accuracy than the unfragmented scheme. Even the naive scheme (in fragmented k -anonymity) was often able to perform better than the unfragmented approach in spite of its relative lack of sophistication in performing the class distortions. The difference in F-measure becomes even more drastic in case of ℓ -diversity and the fragmented anonymization achieves up to 28% improvement over unfragmented scenario.

It is also immediately evident that the trend with in-

²<http://archive.ics.uci.edu/ml>.

Figure 2: Prediction F-measure on *Musk* vs. dimensionality
(a) *J48* classifier (b) *k*-NN classifier

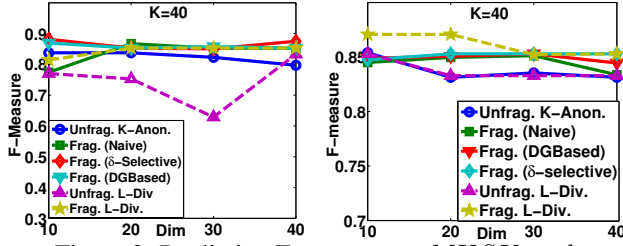
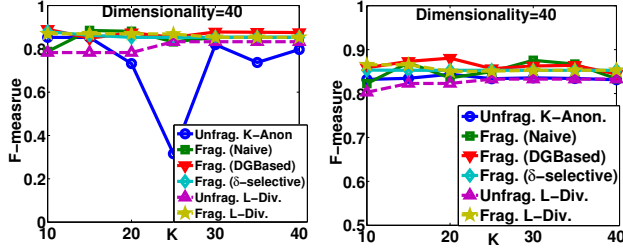


Figure 3: Prediction F-measure on *MUSK* vs. *k*
(a) *J48* classifier (b) *k*-NN classifier



creasing dimensionality is specific to the choice of data set, classifier, and specific approach. In particular, an interesting trend is that the F-measure changes only a little bit with increasing dimensionality in many cases, especially for unfragmented data. This is because the anonymization process in the unfragmented case changes values for many feature attributes to the very general value as the dimensionality grows. This change turns a high-dimensional data set into a data set with very few useful features for the classification. This phenomenon is reflected in the F-measure of unfragmented anonymization shown in Figure 2, which often does not vary much. In fact, only 6 feature attributes in *Musk* played a significant role in the classification. As a result, the F-measure does not vary too much with increasing dimensionality. Besides, as the equivalence classes have different class labels in the case of ℓ -diversity, the precision and recall degrade dramatically and cause the F-measure to be very low for unfragmented ℓ -diversity.

The effect of k on classification F-measure is illustrated in Figure 3. The fragmented anonymization reveals a prominent improvement of up to 54% compared to the unfragmented anonymization. Normally, we expect the prediction F-measure to decline with increasing values of k . While this was often the case, there were also a few cases, where it has an unexpected rise. This trend has also sometimes been observed in earlier work, and is a result of the aggregation effects of the anonymization procedures (sometimes) removing the noisy artifacts in the data.

7 Conclusions and Summary

This paper presents a method for fragmentation-based anonymization for high-dimensional data. While the curse of dimensionality is a fundamental theoretical barrier, it is often possible to obtain effective results in practice. This paper uses fragmentation as a general purpose methodology to improve the effectiveness of any off-the-shelf algorithm for

the anonymization process. Experimental results show significant improvements of the utility of the data after the fragmentation process. This meta-algorithm approach is fairly general and has the potential to be extended to a wider variety of scenarios and privacy models and workloads. This will be the focus of our future work.

References

- [1] R. Agrawal and R. Srikant. Privacy-preserving data mining, *SIGMOD*, 2000.
- [2] C. C. Aggarwal. On k -anonymity and the curse of dimensionality, *VLDB*, 2005.
- [3] C. C. Aggarwal. On randomization, public information, and the curse of dimensionality, *ICDE*, 2007.
- [4] C. C. Aggarwal. Privacy and the dimensionality curse, *Privacy Preserving Data Mining: Models and Algorithms*, Springer, 2008.
- [5] S. Agrawal and J. Haritsa. A framework for high accuracy privacy-preserving data mining, *ICDE*, 2005.
- [6] C. C. Aggarwal, P. S. Yu. Privacy preserving data mining: models and algorithms, Springer, 2008.
- [7] C. Chow and M. Mokbel. Trajectory privacy in location-based services and data publication, *SIGKDD Explorations*, 2011.
- [8] V. Ciriani, S. Capitani Di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati. Combining fragmentation and encryption to protect privacy in data storage, *ACM TOISS*, pp. 1–33, 2010.
- [9] C. Ding and H. Peng. Minimum redundancy feature selection from microarray gene expression data, *CSB*, 2003.
- [10] V. Ganapathy, D. Thomas, T. Feder, H. Garcia-Molina, and R. Motwani. Distributing data for secure database services, *PAIS workshop*, 2011.
- [11] G. Ghinita, P. Karras, P. Kalnis, and N. Mamoulis. Fast data anonymization with low information loss, *VLDB*, 2007.
- [12] G. Ghinita, Y. Tao, and P. Kalnis. On the anonymization of sparse high-dimensional data, *ICDE*, 2008.
- [13] V. Iyengar. Transforming data to satisfy privacy constraints, *KDD*, 2002.
- [14] D. Kifer. Attacks on privacy and deFinetti’s theorem, *SIGMOD*, 2009.
- [15] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Mondrian multidimensional k -anonymity, *ICDE*, 2006.
- [16] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Workload-aware anonymization, *KDD*, 2006.
- [17] K. LeFevre, D. DeWitt, and R. Ramakrishnan. Incognito: efficient full-domain k -anonymity, *SIGMOD*, 2005.
- [18] F. Li, J. Sun, S. Papadimitriou, G. Mihaila, and I. Stanoi. Hiding in the crowd: privacy preservation on evolving streams through correlation tracking, *ICDE*, 2007.
- [19] N. Li, T. Li, S. Venkatasubramanian. t -closeness: Privacy beyond k -anonymity and ℓ -diversity, *ICDE*, 2007.
- [20] T. Li, N. Li, J. Zhang, and I. Molloy. Slicing: A new approach to privacy preserving data publishing, *IEEE TKDE*, 2012.
- [21] H. Liu and H. Motoda. Computational methods for feature selection, Chapman and Hall/CRC data mining and knowledge discovery series, 2007.
- [22] A. Machanavajjhala, J. Gehrke, D. Kifer, M. Venkatasubramanian. ℓ -diversity: Privacy beyond k -anonymity, *ICDE*, 2006.
- [23] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, *IEEE TPAMI*, 2005.
- [24] P. Samarati and L. Sweeney. Generalizing data to provide anonymity when disclosing information, *PODS*, 1998.
- [25] J. Vaidya and C. Clifton. Privacy-preserving association rule mining in vertically partitioned data, *KDD*, 2002.
- [26] X. Xiao and Y. Tao. Anatomy: simple and effective privacy preservation, *VLDB*, 2006.
- [27] W. Wong, N. Mamoulis, and D. Cheung. Non-homogeneous generalization in privacy preserving data publishing, *SIGMOD*, 2010.
- [28] M. Xue, P. Karras, C. Raissi, J. Vaidya, and K. Tan. Anonymizing set-valued data by nonreciprocal recoding, *KDD*, 2012.
- [29] M. Terrovitis, N. Mamoulis, and P. Kalnis. Privacy-preserving anonymization of set-valued data, *VLDB*, 2008.
- [30] Y. Xu, K. Wang, A.W. Fu, and P.S. Yu. Anonymizing transaction databases for publication, *KDD*, 2008.
- [31] N. Mohammad, B. Fung, P. Hung, and C. Lee. Anonymizing healthcare data: a case study on the blood transfusion service, *KDD*, 2009.
- [32] M. Ercan Nergiz, M. Atzori, and C. Clifton. Hiding the presence of individuals from shared databases, *SIGMOD*, 2007.
- [33] D. Kifer and J. Gehrke. Injecting utility into anonymized datasets, *SIGMOD*, 2006.

Supplementary Materials

Appendix A. Pseudocodes

Algorithm 1 depicts how a high-dimensional data set can be broken down into fragments.

Algorithm 1 Fragment Construction Algorithm

```

1: //  $n$  denotes the number of features
2:  $MI$ : a  $(n + 1)$  by  $(n + 1)$  matrix storing the mutual
   information (between the features and the class attribute)
3:  $Fragment1$ ,  $Fragment2$ ,  $nonAssignedFs$  = empty
4: Select two features having the maximum mutual infor-
   mation as two seeds,  $seed1$  and  $seed2$ .
5:  $Fragment1.add(seed1)$ 
6:  $Fragment2.add(seed2)$ 
7: Add the rest of features to  $nonAssignedFs$ 
8: while ( $nonAssignedFs$  not empty)
9:    $bestF, bestFragment$  = empty
10:   $maxContribution$  = 0
11:  foreach  $a_i$  in  $nonAssignedFs$ 
12:    foreach  $f$  in  $Fragment1, Fragment2$ 
13:      if ( $contribution(a_i, f) > maxContribution$ )
14:         $bestF = a_i$ 
15:         $bestFragment = f$ 
16:       $maxContribution = contribution(a_i, f)$ 
17:  Add  $bestF$  to  $f$ 
18:  Remove  $bestF$  from  $nonAssignedFs$ 
19: Add the class attribute to  $Fragment1$  and  $Fragment2$ 

```

Algorithm 2 CreateDependencyGraph($set-of-all-EQ$)

```

1: mark all equivalence classes in  $set-of-all-EQ$  as un-
   visited
2:  $c-id = 1$ 
3:  $toBeProcessed$  = empty
4:  $root$  =  $random(set-of-all-EQ)$  // choose an unvisited
   equivalence class randomly
5:  $toBeProcessed.enqueue(root)$ 
6: while ( $toBeProcessed$  not empty)
7:    $current-node = toBeProcessed.dequeue()$ 
8:    $visit(current-node)$ 
9:   foreach  $EQ$  in  $dependant(current-node)$ 
10:    if ( $EQ$  not visited)
11:      draw an edge from  $current-node$  to  $EQ$  in
12:      component with  $id=c-id$ 
13:       $toBeProcessed.enqueue(EQ)$ 
14: if (any un-visited equivalence class left)
15:    $c-id++$ 
16:   go to line 4

```

Algorithm 2 takes the set of all equivalence classes in a given fragmentation, and constructs the *dependency graph*.

At the end of this process, the variable $c-id$ indicates the number of components with no dependency (connection). The subroutine *dependant* takes an equivalence class eq in either F_1 or F_2 , and returns the set of equivalence classes in the other fragment that can be joined by eq .

Algorithm 3 shows the procedure for enforcing the condition in Theorem 4.1 in a *dependency graph*. This algorithm is invoked for each component the *dependency graph* in order to ensure non-identifiability. In Algorithm 3, the subroutine *change* (line 9) changes a class value having the lowest frequency in its equivalence class to the majority class value in that equivalence class. However, this change might affect the previously satisfied k -anonymity non-reconstructability condition between node un and its *visited* neighbors. Thus, lines 10-14 re-check the previously satisfied reconstructability condition, and if needed, the subroutine *change* is called as necessary in order to re-satisfy it.

Algorithm 3 DGBE($d-graph$)

```

1:  $processing-queue$  = empty
2: mark all nodes in  $d-graph$  as unvisited
3:  $processing-queue.enqueue(random(d-graph))$ 
4: while ( $processing-queue$  not empty)
5:    $current-node = processing-queue.dequeue()$ 
6:   foreach  $un$  in  $unvisited-neighbors(current-node)$ 
7:     while (( $current-node$  and  $un$ ) not satisfy the
8:        $k$ -anonymity non-reconstructability condition)
9:        $change(un)$ 
10:    if (any previously-satisfied  $k$ -anonymity
11:      non-reconstructability condition violates
12:      between  $un$  and visited- neighbors( $un$ ))
13:      call  $change(un)$  until there is no violation
14:      between  $un$  and visited-neighbors( $un$ )
15:     $visit(current-node)$ 
16:     $processing-queue.enqueue(un)$ 

```

Algorithm 4 shows how δ -selectivity can be enforced on a fragmentation \mathcal{F} . This algorithm is very similar to Algorithm 3. It is worth noting that the class values of equivalence classes in the dependency graph ($d-graph$) are at the equivalence class-level. Subroutine *change* simply removes a class value in an equivalence class. In order to reduce the class value distortion, the class value having the lowest frequency before converting the class values into equivalence class-level must be removed.

Appendix B. Proofs

THEOREM 4.1. (K -ANONYMITY NON-RECONSTRUCTABILITY CONDITION). *The condition for a given fragmentation $\mathcal{F} = \{F_1, F_2\}$ which satisfies the fragment k -anonymity condition, and fragments*

Algorithm 4 δ -selectivity(d -graph, δ)

```

1: processing-queue = empty
2: mark all nodes in d-graph as unvisited
3: processing-queue.enqueue(random(d-graph))
4: while (processing-queue not empty)
5:   current-node = processing-queue.dequeue()
6:   foreach un in unvisited-neighbors(current-node)
7:     while ( $\eta(\text{un}, \text{current-node}) < \delta$ )
8:       change(un)
9:       if ( $\eta$  between un and any vn  $\in$  visited-neighbors(un) becomes less than  $\delta$ )
10:        call change(un) until  $\eta(\text{un}, \text{vn})$  becomes
11:        greater
12:        than  $\delta$ 
13:      visit(current-node)
14:    processing-queue.enqueue(un)

```

$F_1 = \{EQ_{11}, EQ_{12}, \dots, EQ_{1n}\}$ and $F_2 = \{EQ_{21}, EQ_{22}, \dots, EQ_{2m}\}$, to be non-reconstructible is that one of the following must be true for each joined pair EQ_{1i}, EQ_{2j} :

- $\sum_{c \in C_{1i} \cap C_{2j}} \text{freq}(c, EQ_{1i}) \times \text{freq}(c, EQ_{2j}) = 0$
- $\sum_{c \in C_{1i} \cap C_{2j}} \text{freq}(c, EQ_{1i}) \times \text{freq}(c, EQ_{2j}) \geq k$

Proof. According to Definition 4.1, \mathcal{F} is non-reconstructible if and only if the relation $F_1 \bowtie F_2$ satisfies k -anonymity.

Joining two equivalence classes EQ_{1i} and EQ_{2j} generates tuples with the same value for their quasi-identifiers. If the number of generated tuples is greater than k , joining EQ_{1i} and EQ_{2j} does not violate the k -anonymity property. For any given equivalence classes EQ_{1i} and EQ_{2j} , one of the following conditions is true:

- $CLS = C_{1i} \cap C_{2j} = \emptyset$.
- $CLS = C_{1i} \cap C_{2j} = \{c_1, c_2, \dots, c_w\}$.

If the former condition is true, EQ_{1i} and EQ_{2j} do not have any common values to be joined on. Therefore, we have:

$$\sum_{c \in \emptyset} \text{freq}(c, EQ_{1i}) \times \text{freq}(c, EQ_{2j}) = 0$$

If the latter condition is true, the joining of EQ_{1i} and EQ_{2j} will result in $\sum_{c \in CLS} \text{freq}(c, EQ_{1i}) \times \text{freq}(c, EQ_{2j})$ tuples. Thus, the result of joining EQ_{1i} and EQ_{2j} is k -anonymous if the number of resulting tuples is at least k . Therefore, in either case the resulting fragmentation \mathcal{F} is k -anonymity non-reconstructible.

THEOREM 5.1. *The data set resulting from joining chunks CK_{s1} and CK_{s2} neither violates k -anonymity nor ℓ -diversity.*

Proof. To prove this theorem, we discriminate between different types of ℓ -diversity, and investigate each model individually.

Distinct ℓ -diversity: The joining of CK_{s1} and CK_{s2} is performed based on joining equivalence classes EQ_{i1}^s and EQ_{j2}^s . According to the Step 4 of fragmentation-based ℓ -diversity, EQ_{i1}^s and EQ_{j2}^s have been constructed in such a way that have the same set (of size l_s) of class values with frequencies $\{f_1^x, f_2^x, \dots, f_{l_s}^x\}$ and $\{f_1^y, f_2^y, \dots, f_{l_s}^y\}$, respectively. Since both EQ_{i1}^s and EQ_{j2}^s satisfy the k -anonymity, we have:

$$\begin{aligned} 1) |EQ_{i1}^s| &= f_1^1 + f_2^1 + \dots + f_{l_s}^1 \geq k \\ 2) |EQ_{j2}^s| &= f_1^2 + f_2^2 + \dots + f_{l_s}^2 \geq k \end{aligned}$$

Evidently, joining EQ_{i1}^s and EQ_{j2}^s are based on the common sensitive values, thus it results in a data set of size $f_1^1 \times f_1^2 + f_2^1 \times f_2^2 + \dots + f_{l_s}^1 \times f_{l_s}^2$ having exactly l_s sensitive values. Therefore, the data set resulting from this join satisfies the distinct ℓ -diversity. Since joining each pair of arbitrary equivalence classes EQ_{i1}^s and EQ_{j2}^s satisfies the distinct ℓ -diversity, joining CK_{s1} and CK_{s2} also satisfies this privacy requirement.

Entropy (or recursive) ℓ -diversity: This part is a bit trickier. Joining each pair of equivalence classes EQ_{i1}^s and EQ_{j2}^s will result in a data set with at least k tuples (with the same reasoning as distinct ℓ -diversity), but most likely different level of diversity than l_s (it can be lower or higher than l_s). This might be considered as a privacy violation. However, joining all possible pairs of equivalence classes EQ_{i1}^s and EQ_{j2}^s generates a data set consisting of both real (all tuples in S_s) and fake tuples. There is an important observation here. If the attacker has enough background knowledge to rule out the faked tuples, the diversity level of real tuples are exactly equal to l_s . Hence, this is not a privacy violation.

Appendix C. Enforcement via δ -selectivity

I. Tuple-level vs. equivalence class-level class value publishing

Table 3 exemplifies the difference between tuple-level and equivalence class-level publishing for a 5-anonymized fragment. As Table 3b illustrates, publishing the class values in the equivalence class-level usually results in an equivalence class with some *ambiguous slots* for its class values. For example, two ambiguous slots are illustrated in Table 3b. The only case in which publishing at the equivalence class-level does not result in an ambiguous slot is when the class values are unique in the equivalence class. As no further information is published regarding the frequency of the class values in each equivalence class, any class value available in the equivalence class can be placed in the ambiguous slots.

Table 3: Tuple-level vs. equivalence class-level class value publishing

(a) Tuple-level class value publishing for $k = 5$.

23	M	flu
23	M	pneumonia
23	M	dyspepsia
23	M	pneumonia
23	M	flu

(b) Equivalence class-level class value publishing for $k = 5$.

23	M	flu pneumonia dyspepsia pneumonia flu
23	M	
23	M	
23	M	
23	M	

The ambiguous slots in an equivalence class EQ_{ij} , published at the equivalence class-level, can take any of the class values in C_{ij} . In other words, different versions for EQ_{ij} can be assumed. Table 4 illustrates two possible versions for the equivalence class in Table 3b.

Table 4: Two possible versions of Table 3b

(a) Version 1

23	M	flu
23	M	flu
23	M	dyspepsia
23	M	pneumonia
23	M	flu

(b) Version 2

23	M	pneumonia
23	M	flu
23	M	dyspepsia
23	M	pneumonia
23	M	dyspepsia

II. A subtle way to find η

Given two equivalence classes EQ_{1i} and EQ_{2j} whose class values are published in equivalence class-level and have at least one class value in common, we find the number of tuples resulted from their join which is referred to as *equijoin selectivity* in the literature. As $|EQ_{1i}|$ and $|C_{1i}|$ refer to the number of tuples and the class values (in equivalence class-level) in EQ_{1i} , the number of ambiguous slots is equal to $|EQ_{1i}| - |C_{1i}|$. The set $\{V(EQ_{1i})\}$ represents different versions (or instantiations) of EQ_{1i} . Each member of this set is a combination of $|EQ_{1i}|$ tuples in which there exist at least one tuple per each class value in C_{1i} . The size of this set is equal to $\binom{(|EQ_{1i}| - |C_{1i}|) + |C_{1i}| - 1}{(|EQ_{1i}| - |C_{1i}|)}$.

There exist $|\{V(EQ_{1i})\}| \times |\{V(EQ_{2j})\}|$ different ways to join EQ_{1i} and EQ_{2j} . Among all possible joins, those generating k tuples (or more) are referred to as *k-anonymity-preserving equijoins*.

In η formula, calculating the denominator is straightforward by multiplying $|V(EQ_{1i})|$ and $|V(EQ_{2j})|$. However, computing the numerator is more challenging. The simplest way is perform all joins, and count those which satisfy the minimum cardinality of k . A more subtle way is to consider the conditions (constraints) that exist on the number of class

values in each equivalence class and map the problem as follows:

The number of pairs (A, B) of integer vectors $A = (a_1, a_2, \dots, a_c)$ and $B = (b_1, b_2, \dots, b_c)$ in which a_d and b_d depict the frequency of d^{th} common class value in $V(EQ_{1i})$ and $V(EQ_{2j})$ respectively, such that the following conditions are satisfied:

- condition 1: $\sum a_d \times b_d \geq k$
 - condition 2: $1 \leq a_d \leq (1 + |EQ_{1i}| - |C_{1i}|)$
 - condition 3: $1 \leq b_d \leq (1 + |EQ_{2j}| - |C_{2j}|)$
 - condition 4: $\sum a_d \leq |CLS| + |EQ_{1i}| - |C_{1i}|$
 - condition 5: $\sum b_d \leq |CLS| + |EQ_{2j}| - |C_{2j}|$
- where $CLS = C_{1i} \cap C_{2j}$.

Appendix D. Experiments

I. Performance measures

We utilized two metrics to evaluate the effectiveness of our proposed method. The first one is the widely-used metric known as *information loss*, which captures the total amount of lost information due to generalization. In fact, this metric shows the usefulness of data *on a per-attribute basis* for general workloads. For an anonymized data set with n tuples and m attributes, the information loss I is computed as follows:

$$I = \sum_{i=1}^n \sum_{j=1}^m \frac{|upper_{ij} - lower_{ij}|}{n \cdot m \cdot |max_j - min_j|}$$

Here, $lower_{ij}$ and $upper_{ij}$ represent the lower and upper bounds of attribute j in tuple i after generalization, and min_j and max_j represent the maximum and minimum values taken by attribute j over all records. Note that the computation in the fragmented and unfragmented case is not different, as long as all the different attributes in the different fragments are used. This is the most direct measure of the data quality, since it normalizes the final result by the number of attributes.

To evaluate the utility of the data anonymized by our meta-algorithm, we calculated the weighted F-measure of a classifier trained on the anonymous fragmentation. This metric reflects the goodness of classifier more accurately in case on unevenly distributed test data. Each fragment was used to train the classifier separately. For a given test instance, the different fragments of the training data were trained separately, and the weighted majority label from the different classifiers was reported, where the weight used was score returned by the classifier from each fragment. Each class label was once considered as positive (and once as negative) and the weighted F-measure for each case was calculated by taking into account the fraction of the positive instances. We used J48³ and a k -NN⁴ classifier in Weka with

³J48 is an open source implementation of C4.5 in Java, <http://weka.sourceforge.net/doc/weka/classifiers/trees/J48.html>

⁴<http://weka.sourceforge.net/doc/weka/classifiers/lazy/IBk.html>

Table 5: Class distributions in *Musk* data set

class	train	test
musk	1017	207
non-musk	3983	1867

the default setting. The value of k in k -NN was set to 5. The learning from generalized values was also done by the technique used in [16]. In each case, the decomposition was performed into two fragments.

In addition, the amount of distortion required to the class values (for fragmentation k -anonymity) was measured. Specifically, the number of distorted class values were computed for each of the different techniques. The aim is to show that the amount of distortion required was relatively small.

II. Data set description

Real data set *Musk* from the *UCI Machine Learning Repository*⁵ was used. It contains 7074 instances with 168 numerical feature attributes describing a set of molecules. The goal is to predict whether a molecule is musk or non-musk. In order to test the effects of varying data dimensionality, we chose 10, 20, 30, and 40 features randomly and constructed four versions of *Musk* with different dimensionality. Around 70% of the *Musk* data set was used for training. The distribution of the two classes in the training and test data set is shown in Table 5. Although the data set used in this work is numerical, the proposed meta-algorithm can be used to anonymize categorical values by simply using an off-the-shell anonymization technique capable to anonymize categorical values in Step 2 and 4 of fragmentation k -anonymity and ℓ -diversity, respectively.

⁵<http://archive.ics.uci.edu/ml>.